PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification [6] : H04L 29/06 | A1 | (11) International Publication Number: WO 98/42108

(43) International Publication Date: 24 September 1998 (24.09.98)

(21) International Application Number: PCT/US98/05532

(22) International Filing Date: 19 March 1998 (19.03.98)

(30) Priority Data:
08/821,235 20 March 1997 (20.03.97) US

(71) Applicant: ERICSSON INC. [US/US]; 7001 Development Drive, P.O. Box 13969, Research Triangle Park, NC 27709 (US).

(72) Inventors: GRAHAM, D., Greg; 544 Savannah Avenue, Lynchburg, VA 24502 (US). DOIRON, Timothy, J.; 403 N. Clinton, New Athens, IL 62264 (US). SPIVEY, Anthony; 312-A Gouyer Drive, Madison Heights, VA 24572 (US). SADROZINSKI, Peter; Apartment 11, 2937 Rivermont Avenue, Lynchburg, VA 24503 (US). CROUCHER, Russell; 208 Crestview Drive, Forest, VA 24551 (US).

(74) Agents: MOORE, Stanley, R. et al.; Jenkens & Gilchrist, P.C., Suite 3200, 1445 Ross Avenue, Dallas, TX 75202 (US).

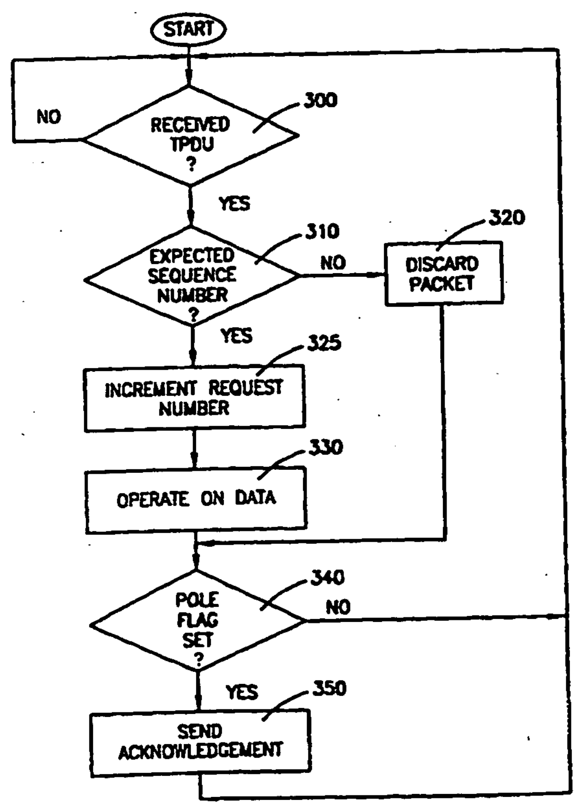
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD FOR IMPLEMENTING A TRANSPORT LAYER PROTOCOL FOR WIRELESS PACKET DATA DELIVERY

(57) Abstract

The present invention is a transport layer protocol for data packet delivery in a wireless environment. The method of the present invention incorporates a transport data protocol structure to implement a modified go-back-n automatic repeat request for data packet delivery. The invention further employs a distinct pair of sequence numbers and request numbers for client and server originated data packet transmissions. Several packets can be transmitted by the sender without the need for each packet to be individually acknowledged by the receiver. After n successive packets have been transmitted, the sender polls the receiver for an acknowledgment. The acknowledgment (350) sent by the receiving party contains the request number equal to the sequence number of the packet that the receiver requests to be sent next (325). This number represents the last packet which was received in order plus one (325). Lastly, the present invention provides a method for the client and server to imply that an acknowledgment (350) has been sent to it even though the acknowledgement was lost in transmission.



METHOD FOR IMPLEMENTING A TRANSPORT LAYER PROTOCOL FOR WIRELESS PACKET DATA DELIVERY

BACKGROUND OF THE INVENTION

Technical Field of the Invention

The present invention pertains in general to a transport layer protocol and more particularly, to a method for implementing a modified go-back-n automatic repeat request protocol in the transport layer for wireless packet data delivery.

Description of Related Art

The International Standards Organization (ISO) has established a seven layer model regarding communication protocols. The forth layer of this model is referred to as the transport layer and is responsible for packet data delivery between a client and a server. To date, the vast majority of communication using packet data delivery has occurred via a wireline link. A standard solution for providing reliable packet data delivery in the wired environment is the use of Transport Control Protocol (TCP). Transport Control Protocol was designed for wired environments characterized by relatively reliable physical connections between communication devices having large memories and computing power.

For various known reasons Transport Control Protocol is unsuited for the harsh environment of wireless packet switched networks. For example, Transport Control Protocol requires a large memory to store both the software for implementing the Transport Control Protocol and the numerous data packets necessary for implementing the Transport Control Protocol's automatic repeat request methodology which allows for the receipt of out-of-order packets. Wireless communication, however, is characterized by the use of radios with small memories and limited computing power as compared to the wired environment. A further problem related to the use of

The present invention provides for a separate and distinct pair of sequence numbers and request numbers for client originated data transmissions and server originated data transmissions. Thus, for client originated data transmissions, the client sequentially numbers transmitted data packets with one set of sequence numbers and the server responds with request numbers equal to the sequence number of the data packet next expected to be sent. Conversely, for server originated data transmissions, the server sequentially numbers transmitted data packets with another set of sequence numbers distinct from client originated transmissions and the server responds with request numbers equal to the sequence number of the data packet next expected to be sent.

The present invention further provides a method for the client and server to imply that an acknowledgment has been sent even though the acknowledgment may have been lost in transmission. The client and server imply an acknowledgment if a subsequent packet is received having an expected sequence number.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the method of the present invention may be acquired by reference to the following Detailed Description when taken in conjunction with the accompanying Drawings wherein:

Figure 1 is a Transport Protocol Data Unit for implementing the present invention;

Figure 2 is a sequence diagram for initiating a communication session;

Figure 3 is a flow diagram for implementing the go-back-n automatic repeat request method of the present invention;

Figure 4 is a sequence diagram for transmitting data from a client to a server without the loss of any data;

Figure 5 is a sequence diagram for transmitting data from a client to a server when a packet of data is lost;

receiver sends an acknowledgment to the sender. The fifth byte of the TPDU 100 comprises either the most significant eight bits of a session ID 181 or an eight bit value representing a retry time 182. During an initialization command sent by a client to a server the session ID has not yet been defined and the fifth byte of the TPDU 100 contains an eight bit value informing the server of the initial retry timeout value. This value has a five second resolution making the initial retry time in seconds equal to five times the value provided in this field. Upon receiving the initialization command, the server sends an acknowledgment to the client containing the most significant eight bits of the session ID 181 in the fifth byte and the least significant eight bits of the session ID 183 contained in the sixth byte of the TPDU 100. The most significant eight bits of the session ID 181 together with the least significant eight bits of the session ID 183 comprise a sixteen bit value identifying the current communication session between the client and the server. All TPDUs transmitted between the client and the server must contain the session identifier, otherwise, the data will be ignored.

The seventh and eighth byte of the TPDU 100 comprises the sequence number 180 and the ninth and tenth byte of the TPDU 100 comprises the request number 190. Every TPDU contains a unique sequence number identifying that particular TPDU 100. When a communication session between a client and a server is set up, numbers based on the current time of day are selected for client originated sequence number 180 and for the server originated sequence number which is communicated to the server via the request number 190. For the duration of the communication session, each TPDU 100 is identified with sequentially increasing sequence numbers 180.

The last two bytes of the TPDU 100 forming the trailer 120 contain a sixteen bit cyclic redundancy check for the TPDU 100.

a previously, correctly received TPDU. If the sequence number 180 of the TPDU does not match the expected sequence number, the receiving party discards the received packet (step 320). If, on the other hand, the sequence number 180 of the TPDU matches the expected sequence number the receiving party increments the expected sequence number (step 325) and performs an operation on the TPDU (step 330) in accordance with the command contained in the TPDU field 140. After the receiving party has operated on the data (step 330), or discarded a packet containing a sequence number 180 other than the expected sequence number (step 320) the receiving party determines whether the pole bit 170 is set on the received TPDU (step 340). If the pole bit 170 is not set the receiving party returns to continue monitoring for further transmitted packets (step 300). If, on the other hand, the pole bit 170 is set, the receiving party sends an acknowledgment (step 350) to the sending party, and then returns to continue monitoring for transmitted packets (step 300). The acknowledgment sent by the receiving party contains the request number 190 equal to the sequence number 180 of the packet that the receiver requests to be sent next. This number represents the last packet which was received in order plus one. The method described in the flow diagram of Figure 3 applies equally both to client and server originated transmissions.

Referring additionally now to Figure 4, there is illustrated a sequence diagram for transmitting data from a client 200 to a server 210 when no data packets have been lost. After a communication session has been established, the client 200 proceeds to transmit data packets to the server 210. The number of packets p which the client 200 transmits to the server 210 before requesting an acknowledgment is set by the user. The client 200 transmits multiple data packets to the server 210 and eventually the client 200 transmits the p-1 data packet 400 having a sequence number 180 of a. Since one

Referring additionally now to Figure 6, there is illustrated a sequence diagram for transmitting data from a server 210 to a client 200. This diagram includes server 210 originated data transmissions and illustrates the distinction between sequence numbers 180 and request numbers 190 associated with client 200 originated and server 210 originated transmissions.

The client 200 initiates a communication session by transmitting an initialization TPDU 600. The initialization TPDU 600 contains a randomly generated sequence number 180 of y for client 200 originated transmissions and a time based request number 190 of z corresponding to server 210 originated transmissions. The server 210 upon receiving the initialization TPDU 600 transmits an acknowledgment TPDU 610 containing a request number 190 of y+1. Upon receiving the acknowledgment TPDU 610 the client 200 transmits a data request TPDU 620 in which the pole bit 170 is set thereby requesting an acknowledgment. Since this TPDU 620 is a client 200 originated TPDU the sequence number 180 is equal to the sequence number y of the previously transmitted TPDU 600 plus 1 or y+1. The data request TPDU 620 also includes a request number 190 of z which is the sequence number 180 which the server 210 will use to begin data transmission. Upon receiving the data request 620 the server 610 transmits an acknowledgment TPDU 630 having a request number 190 of y+2. The server 210 then begins to transmit the requested data with server 210 originated TPDU transmissions. These transmissions have sequence numbers 180 beginning with z as requested by the client. A data TPDU 640 having a sequence number 180 of z and a request number 190 of y+2 is thus transmitted to the client 200 with the pole bit 170 set to request acknowledgment. The data TPDU 640 also contains the client 200 requested data. Upon receiving the data TPDU 640 the client 200 transmits an acknowledgment TPDU 650 having a request number 190 of z+1 equal to the next server 210 originated packet

request number y+2 of the data TPDU 720 contains the appropriate request number 190 and implies that an acknowledgment TPDU 710 has been sent by the server 210. The client 200, therefore, retains the data TPDU 720 and transmits an acknowledgment 730 having a request number 190 equal to z+1 to the server 210. In a similar fashion, this acknowledgment TPDU 730 may be unsuccessfully received by the server 210. Nevertheless, the client 200 transmits another TPDU, which for this example is data TPDU 740 having a sequence number 180 equal to y+2 and a request number 190 equal to z+1. The server 210, following the methodology of the present invention, recognizes that the request number z+1 is the appropriate request number 190 and implies that the client 200 has transmitted the acknowledgment TPDU 730 even though it was not received by the server 210. The stipulation of implying an acknowledgment if a packet is subsequently received which has the expected request number 190 prevents the client 200 and the server 210 from becoming out of synchronization with one another and constantly re-sending data packets to one another.

Unlike Transport Control Protocol, the transport layer protocol of the present invention does not support a selective automatic repeat request methodology and thus, does not support the receipt of out of order packets. In the present invention, packets received out of order are discarded. If, however, a packet is received out of order but contains a request for acknowledgment, an acknowledgment is sent containing the request number equal to the sequence number of the last packet received in order plus one.

Although a preferred embodiment of the method of the present invention has been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it is understood that the invention is not limited to the embodiment disclosed, but is capable of numerous rearrangements, modifications and substitutions

WHAT IS CLAIMED IS:

1. A Transport Protocol Data Unit for a wireless packet data delivery comprising:

a response bit for signifying the Transport Protocol Data Unit as a command or response Transport Protocol Data Unit;

a poll bit for requesting an acknowledgment;

a sequence number field for identifying the Transport Protocol Data Unit; and

a request number field for communicating a sequence number of the Transport Protocol Data Unit requested to be next transmitted.

2. A method for data packet delivery between a sender and a receiver, comprising the steps of:

transmitting a user selected number of data packets, wherein each packet is sequentially numbered and each packet except for a last packet has a poll bit cleared, the last packet having the poll bit set;

receiving the data packets transmitted by the sender;

detecting the sequence number and the poll bit of each received packet;

comparing the sequence number of each received packet against an expected sequence number;

storing data contained within packets when the sequence number of the packet matches the expected sequence number; otherwise

discarding the data packets.

3. A method for data packet delivery between a sender and a receiver, comprising the steps of:

transmitting a user selected number of data packets, wherein each packet is sequentially numbered and each packet except for a last packet has a poll bit cleared, the last packet having the poll bit set;

receiving the data packets transmitted by the sender;

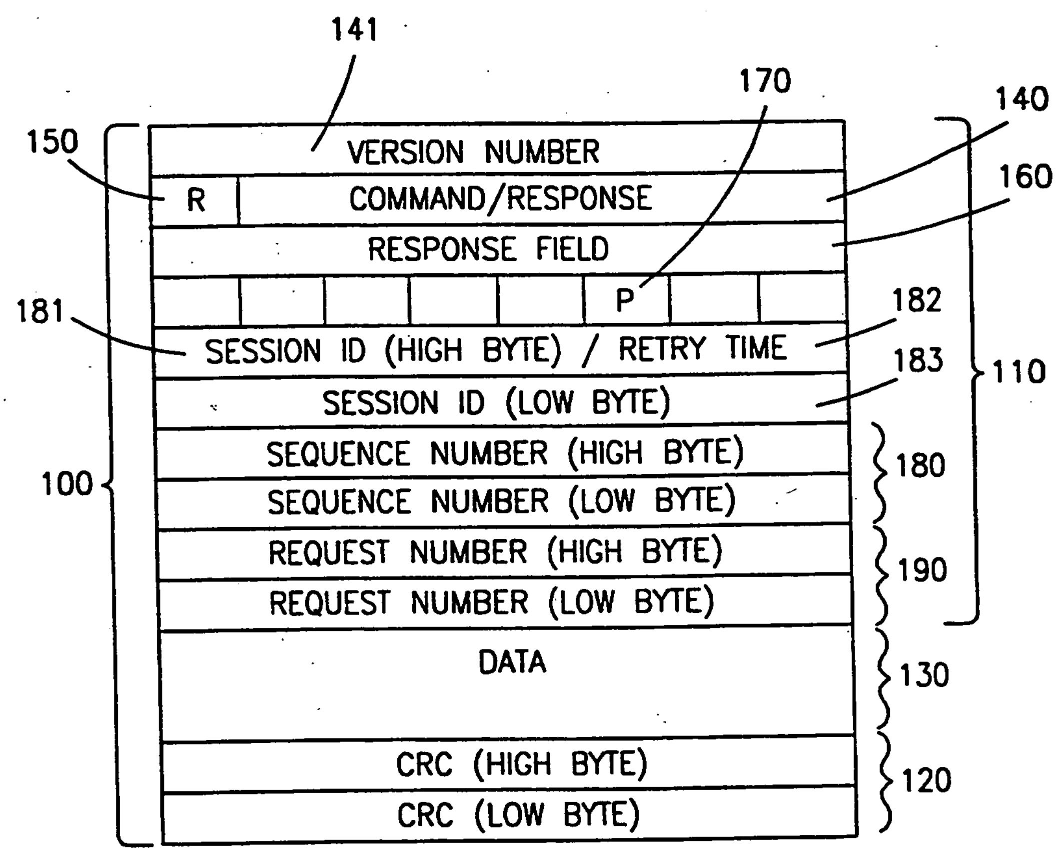

141
170
140
150
VERSION NUMBER
R
COMMAND/RESPONSE
160
RESPONSE FIELD
P
181
182
SESSION ID (HIGH BYTE) / RETRY TIME
183
SESSION ID (LOW BYTE)
110
SEQUENCE NUMBER (HIGH BYTE)
180
100
SEQUENCE NUMBER (LOW BYTE)
REQUEST NUMBER (HIGH BYTE)
190
REQUEST NUMBER (LOW BYTE)
DATA
130
CRC (HIGH BYTE)
120
CRC (LOW BYTE)

FIG. 1

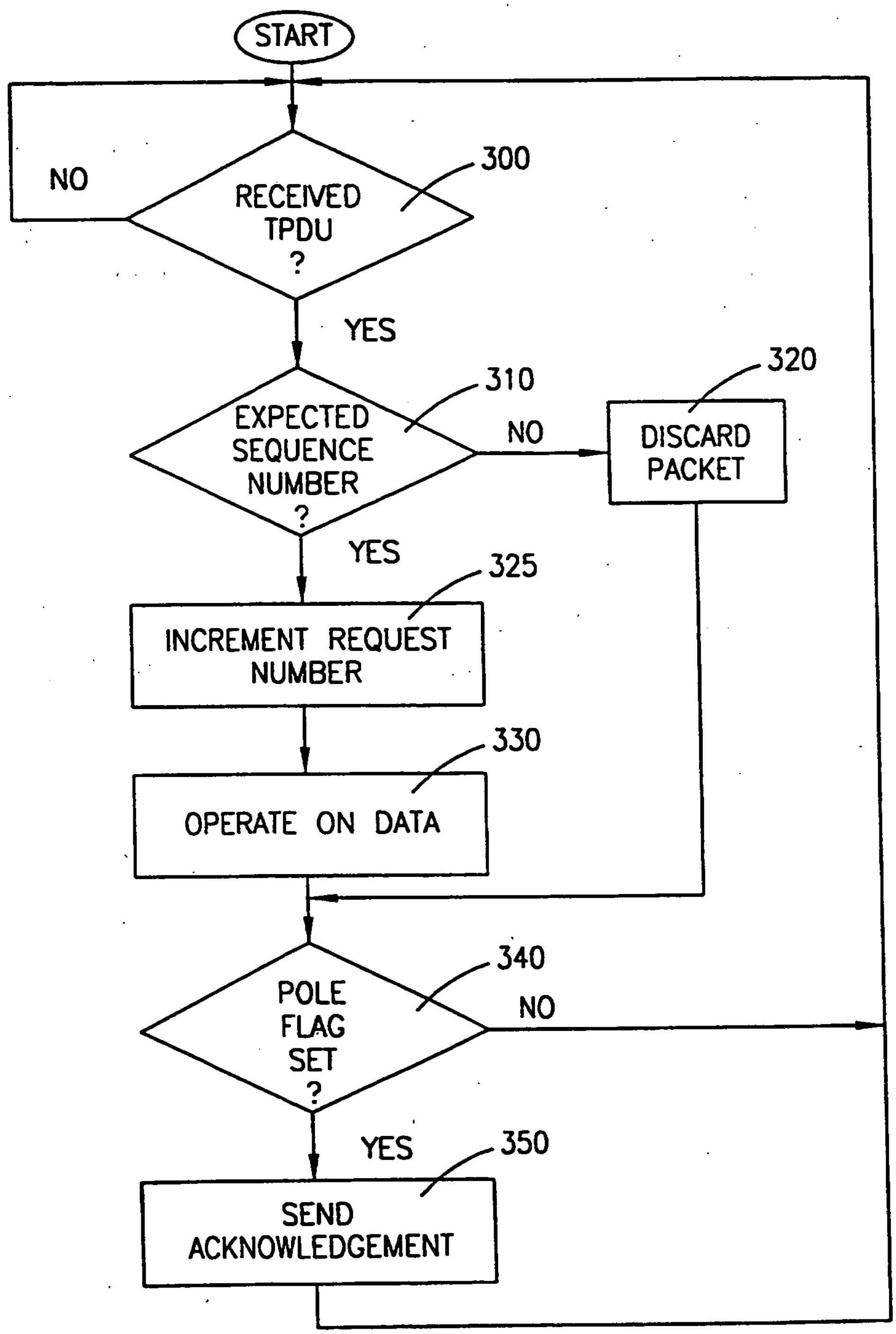

START
NO
RECEIVED TPDU ?
300
YES
EXPECTED SEQUENCE NUMBER ?
310
NO
DISCARD PACKET
320
YES
INCREMENT REQUEST NUMBER
325
OPERATE ON DATA
330
POLE FLAG SET ?
340
NO
YES
SEND ACKNOWLEDGEMENT
350

FIG. 3

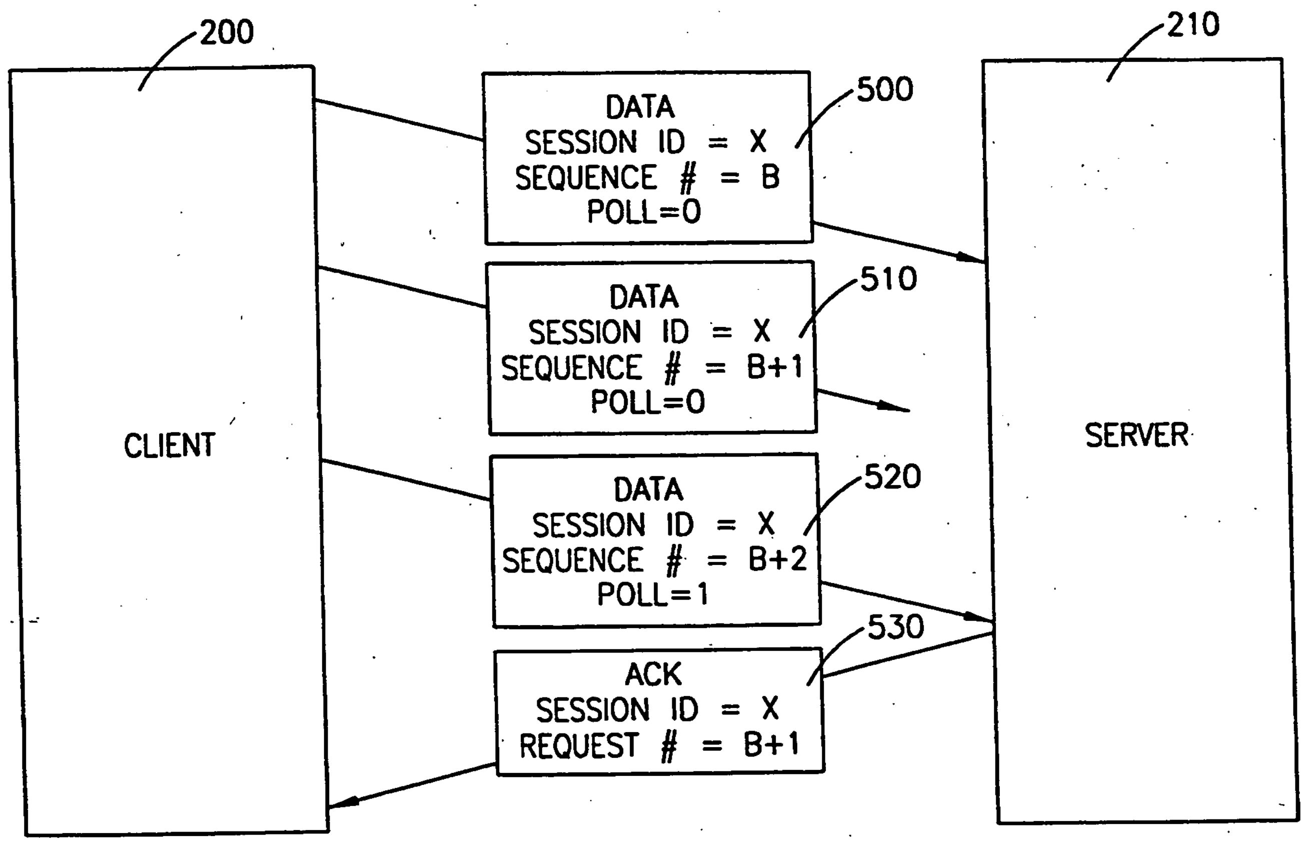

200
210
CLIENT
SERVER
DATA
SESSION ID = X
SEQUENCE # = B
POLL=0
500
DATA
SESSION ID = X
SEQUENCE # = B+1
POLL=0
510
DATA
SESSION ID = X
SEQUENCE # = B+2
POLL=1
520
ACK
SESSION ID = X
REQUEST # = B+1
530

FIG. 5

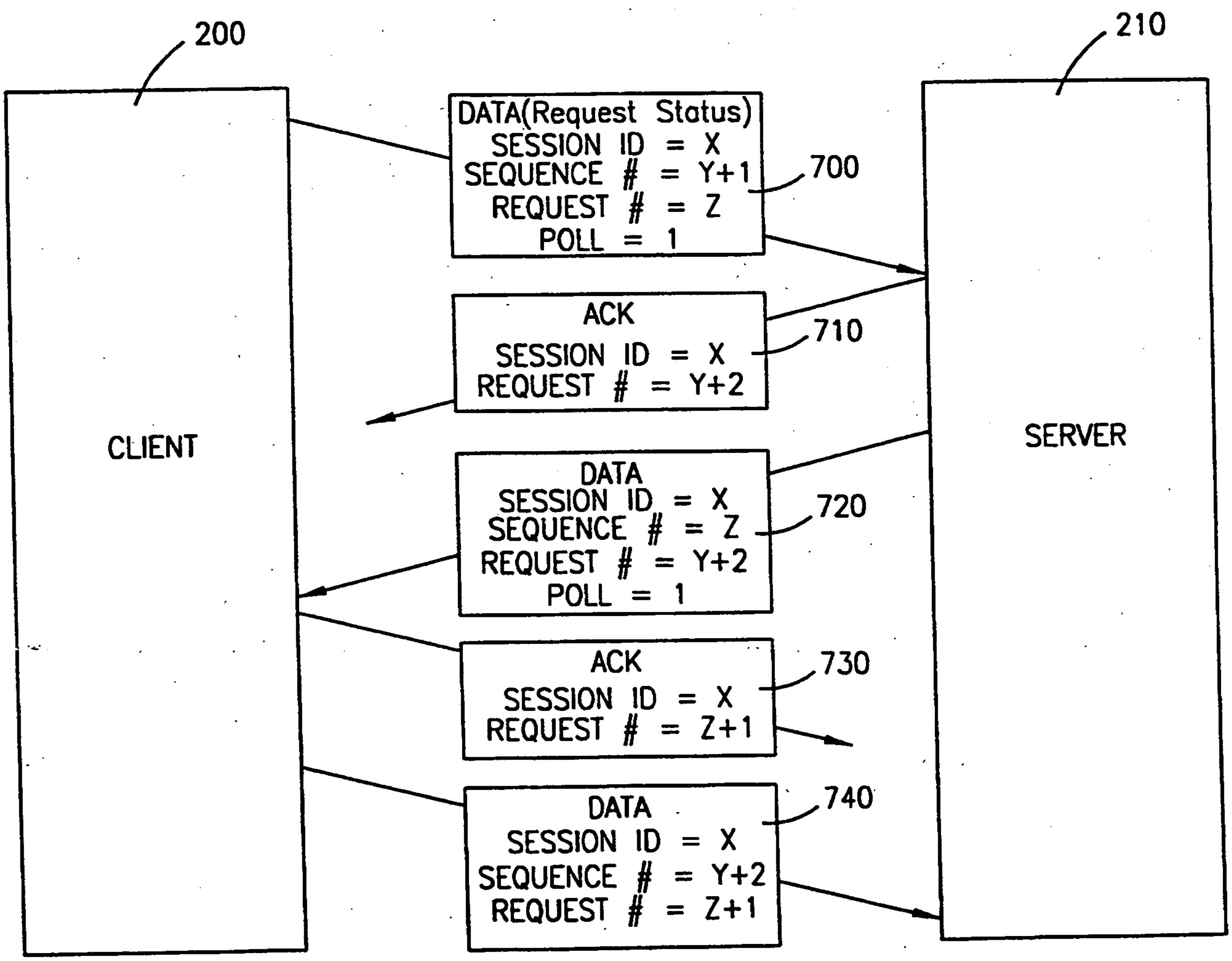

200
210
CLIENT
SERVER
DATA(Request Status)
SESSION ID = X
SEQUENCE # = Y+1
REQUEST # = Z
POLL = 1
700
ACK
SESSION ID = X
REQUEST # = Y+2
710
DATA
SESSION ID = X
SEQUENCE # = Z
REQUEST # = Y+2
POLL = 1
720
ACK
SESSION ID = X
REQUEST # = Z+1
730
DATA
SESSION ID = X
SEQUENCE # = Y+2
REQUEST # = Z+1
740

FIG. 7

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 98/05532

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 748 096 A (XEROX CORP) 11 December 1996 see page 3, line 16 - page 4, line 8 ------	1-3

3

Form PCT/ISA/210 (continuation of second sheet) (July 1992)